Parallel Performance of Structured AMR Calculations Using the SAMRAI Framework

Andrew Wissink

with

Richard Hornung, Scott Kohn, David Hysom, Steve Smith, Noah Elliott, Brian Gunney

Center for Applied Scientific Computing

Lawrence Livermore National Laboratory

August 14, 2002
BlueGene/L Workshop

Center for Applied Scientific Computing

Outline

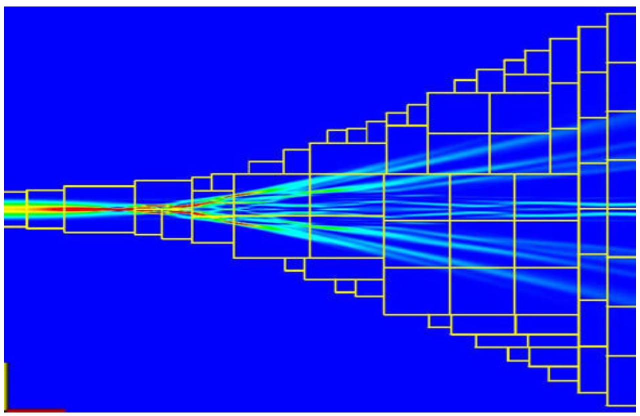
- SAMRAI introduction
- Parallel implementation of SAMR
- Parallel performance measurements
- New algorithms to enhance parallel performance
- Requirements and issues on BG/L

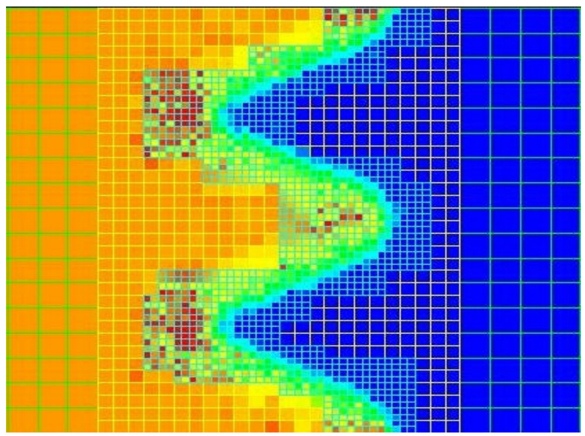
SAMRAI

Structured Adaptive Mesh Refinement Application Infrastructure

- **Object-oriented (C++) software framework for parallel (MPI) adaptive multi-physics applications**
- **Supports applications investigating multi-scale phenomena.**
- **High-level reusable code and algorithms shared across a variety of applications.**

www.llnl.gov/CASC/SAMRAI

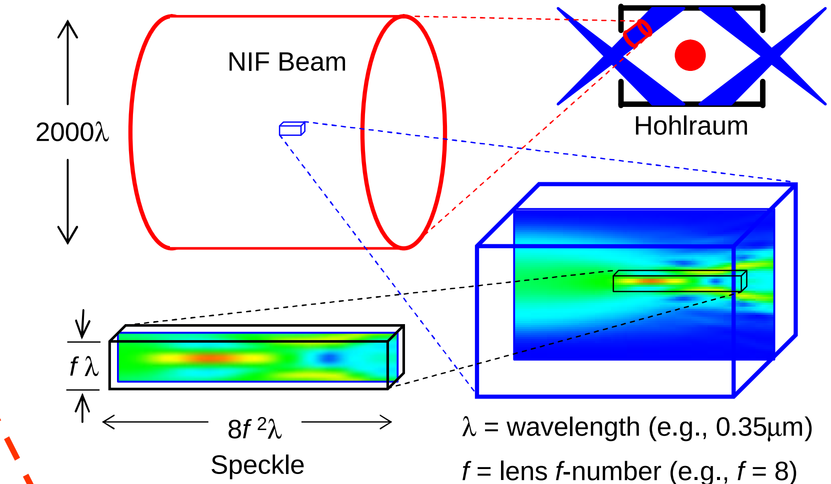




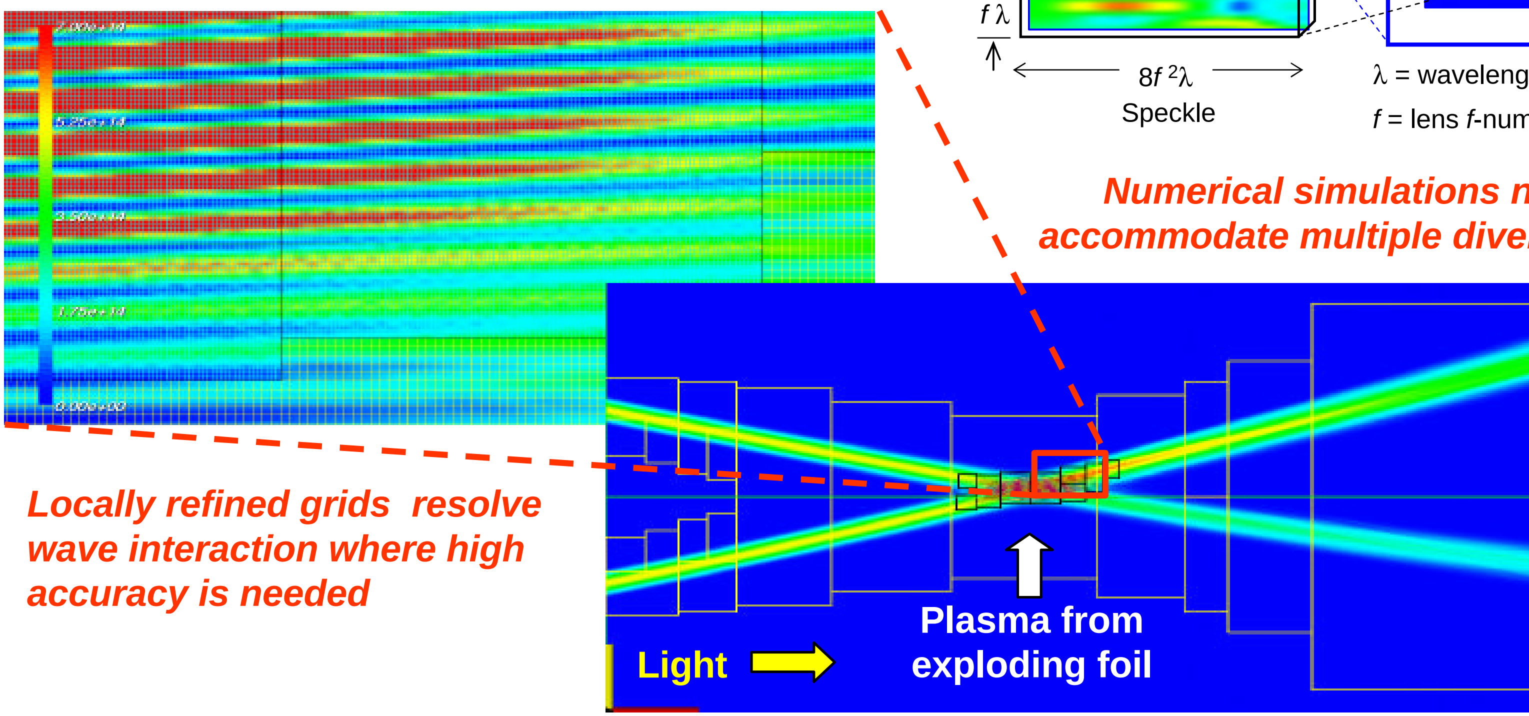
ALPS uses SAMRAI for adaptive laser plasma instability simulation

Understanding instabilities in laser-plasma interactions is critical in the design of plasma physics experiments

Dorr, Garaizar, Hittinger (CASC-LLNL)



Numerical simulations need to accommodate multiple diverse scales

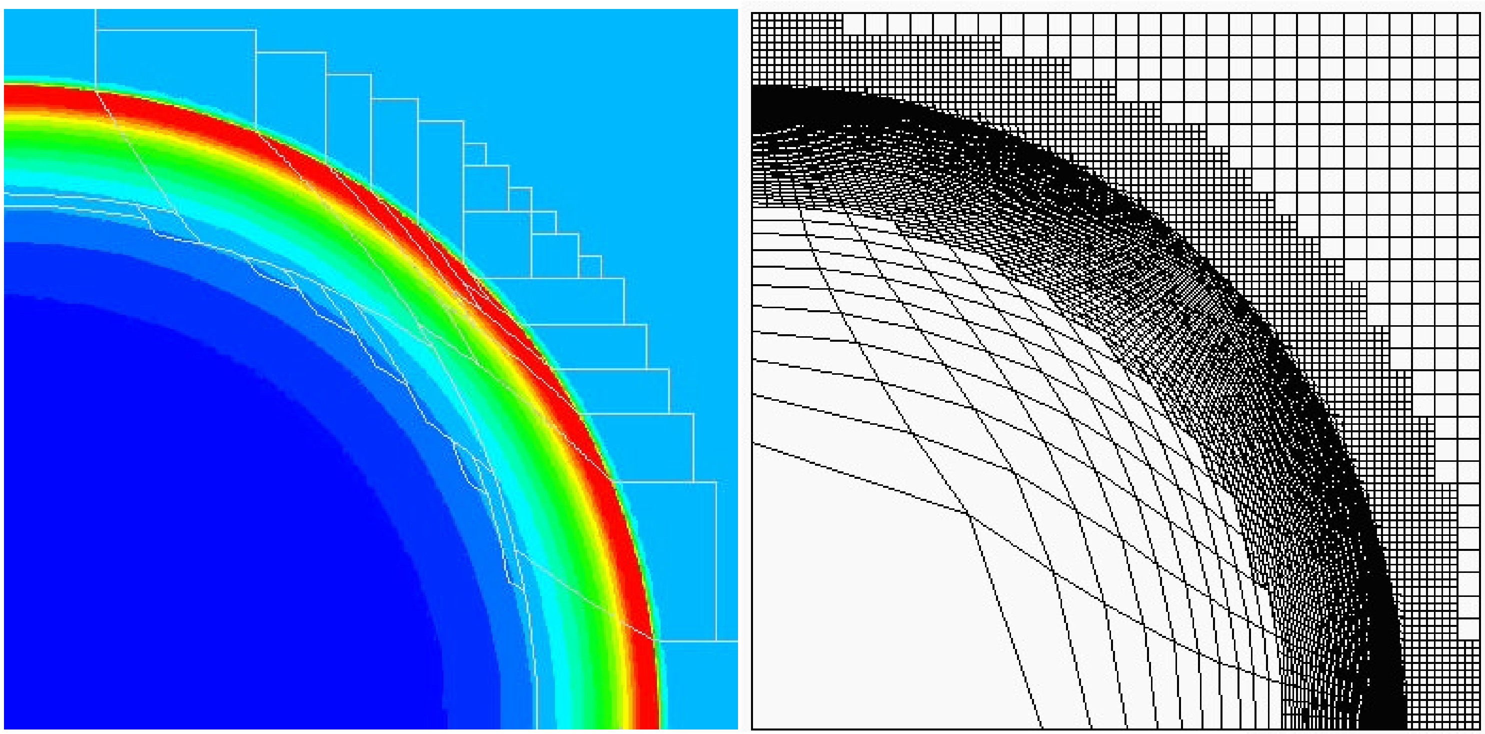


Locally refined grids resolve wave interaction where high accuracy is needed

ALE-AMR couples ALE models with AMR to model shock hydrodynamics

Improve accuracy of ALE simulations by increasing concentration of mesh points around regions of interest

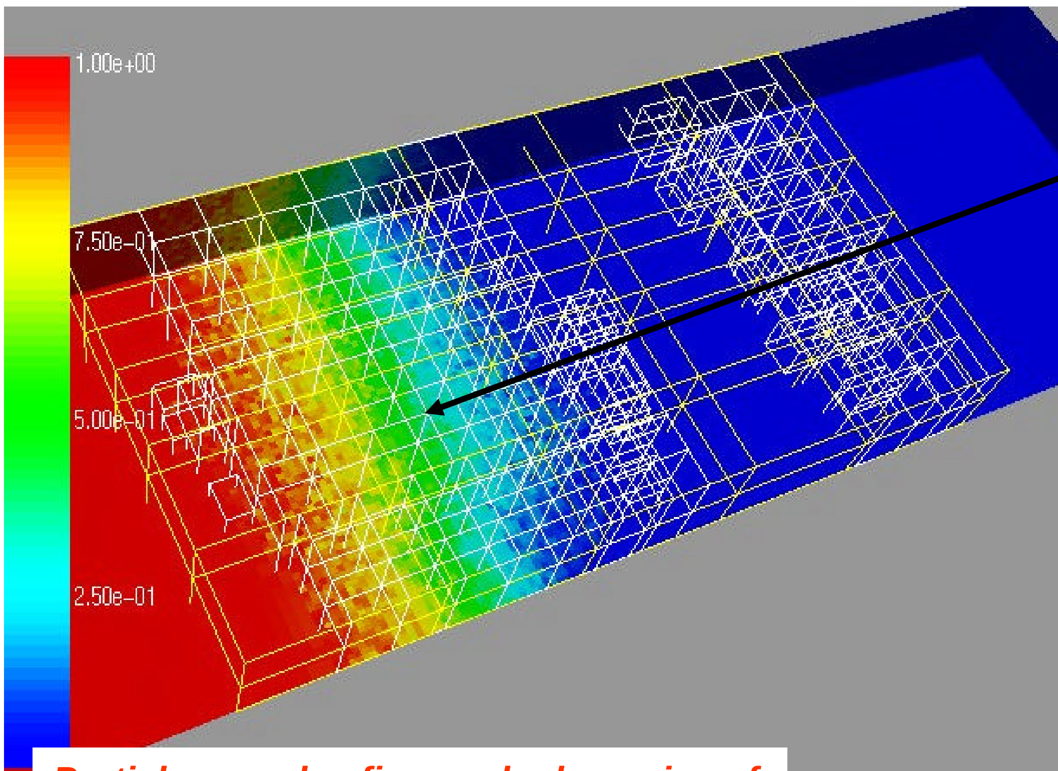
Anderson, Pember, Elliott (CASC-LLNL)



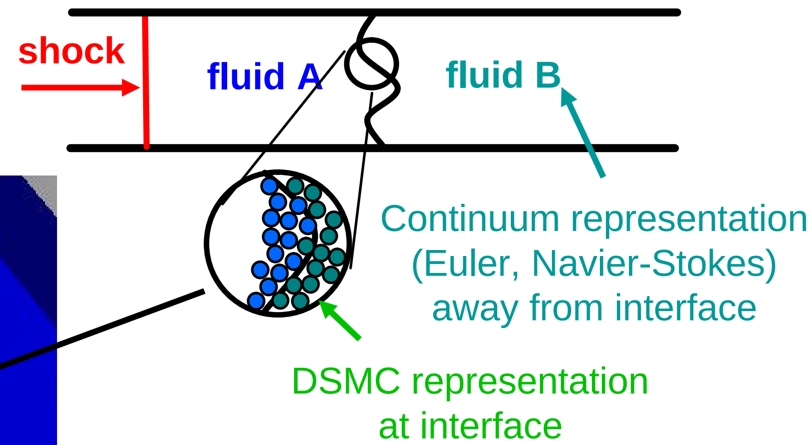
Sedov blast wave density and Lagrangian mesh

Hybrid continuum-DSMC model used to efficiently resolve interface dynamics

Interface instability problems (e.g., Richtmyer-Meshkov) involve coarse-scale hydrodynamic transport and fine-scale molecular diffusion



Particles resolve fine-scale dynamics of mixing region in an adaptive calculation



- **Interface region grows and moves as instability evolves**
- **Standard CFD simulation of turbulent mixing is limited by finest mesh scale**
- **Particle resolve molecular behavior but are too expensive for large domains**

Hornung (CASC), Garcia (SJSU)

SAMRAI provides infrastructure support for a variety of applications

- **Parallel processing support (MPI)**
- **Shared algorithms**
- **Interfaces for SAMR data to solvers (PETSc, PVODE, *hypre*)**
- **Checkpointing & restart support (HDF)**
- **Parallel tools (VAMPIR, TAU)**
- **Current users regularly run on existing large processor systems**

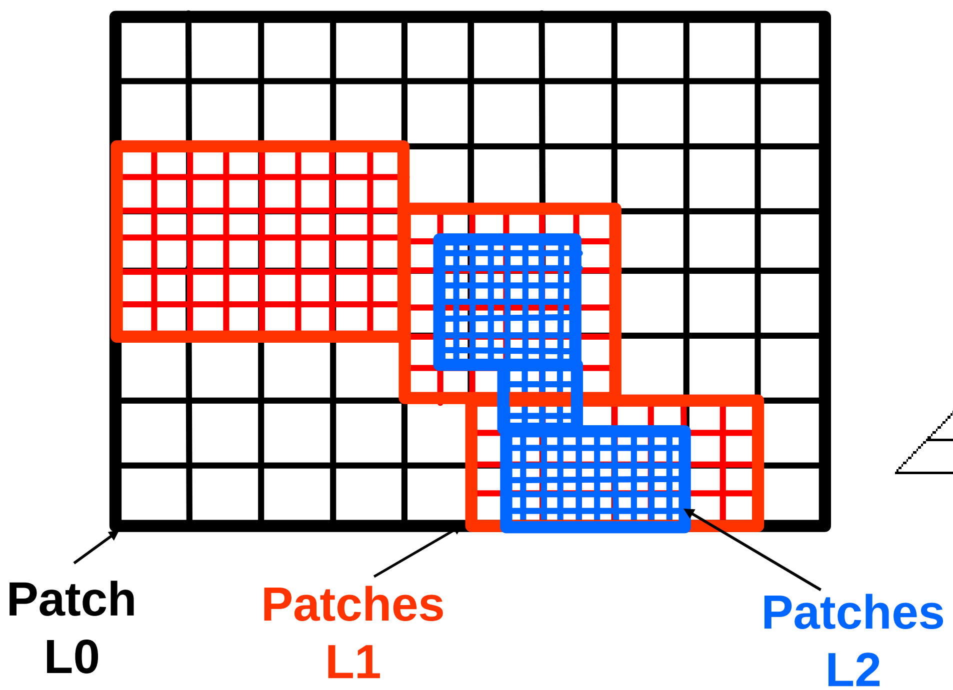
SAMRAI on BG/L → **Applications on BG/L**

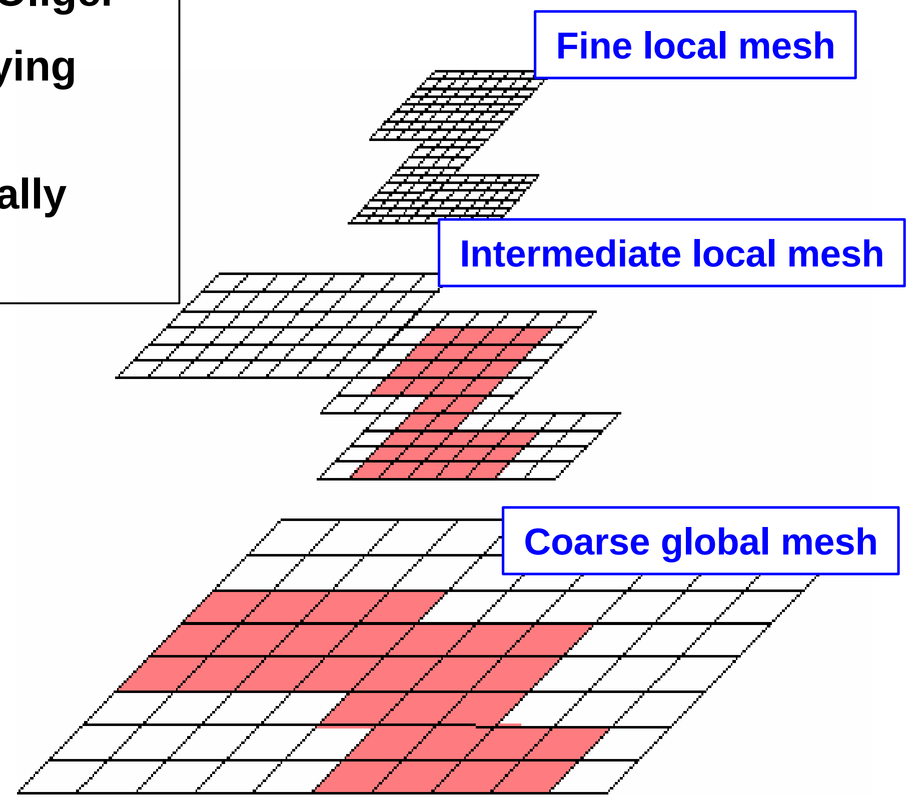
Outline

- SAMRAI introduction
- **Parallel implementation of SAMR**
- Parallel performance measurements
- New algorithms to enhance parallel performance
- Requirements and issues on BG/L

Structured AMR (SAMR) employs a dynamically adaptive "patch" hierarchy

- Based on methods of Berger, Colella, Oliger
- Hierarchy defines nested levels of varying mesh resolution
- Data stored on patches covering logically rectangular index space





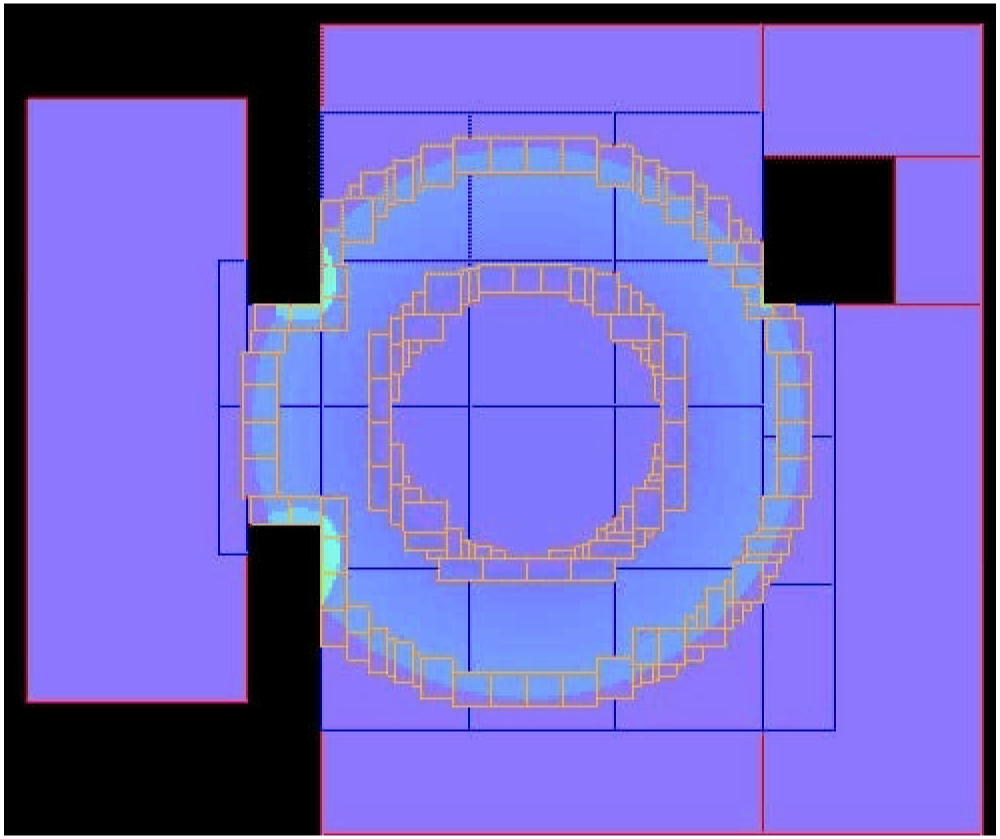
Dynamic mesh adapts to features as solution evolves

Adaptive solution of Euler equations

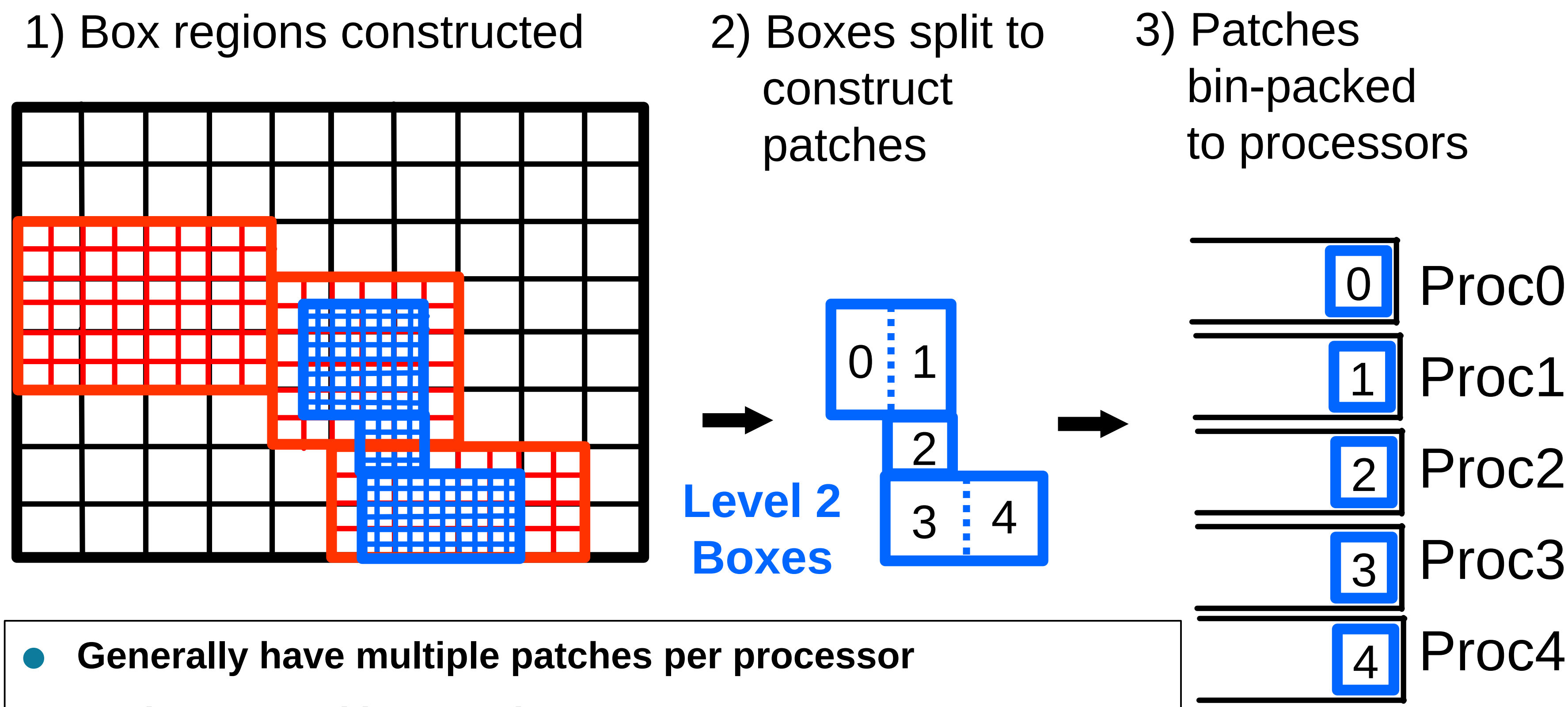
Initial conditions:

inside sphere
density = 8.0
pressure = 40.0

outside sphere
density = 1.0
pressure = 1.0



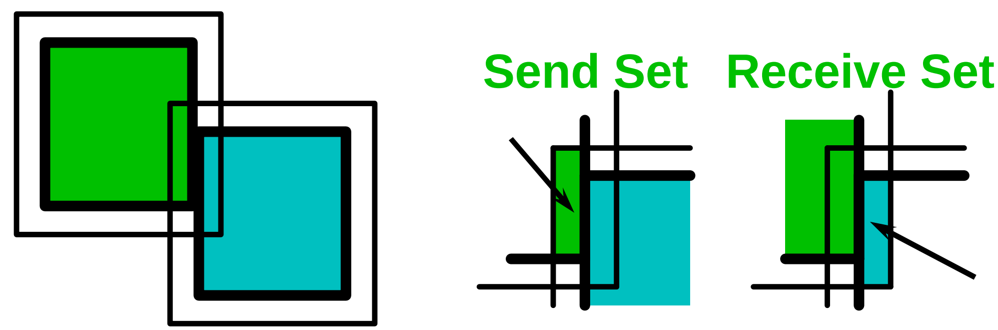
Patches distributed to processors to balance computational workload



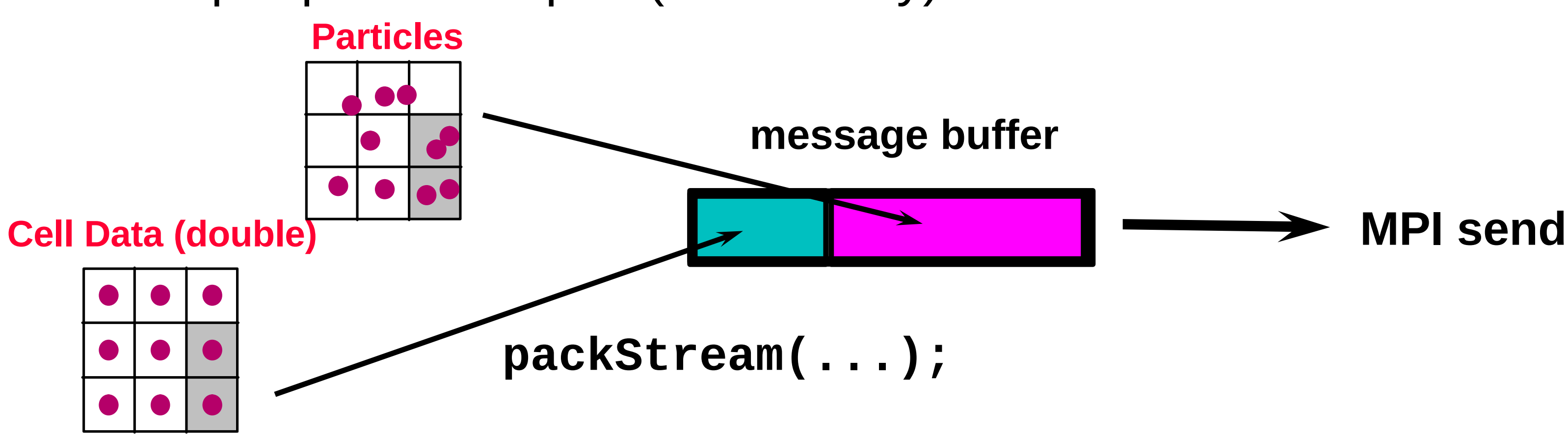
- **Generally have multiple patches per processor**
- **Each level load balanced separately**
- **Spatial bin packing may be used to maintain locality of patches on processors**

Communication schedules create and store data dependencies

- **Amortize cost of creating send/receive sets over multiple communication cycles**



- **Data from various sources packed into single message stream**
  - supports complicated variable-length data
  - one send per processor pair (low latency)



Outline

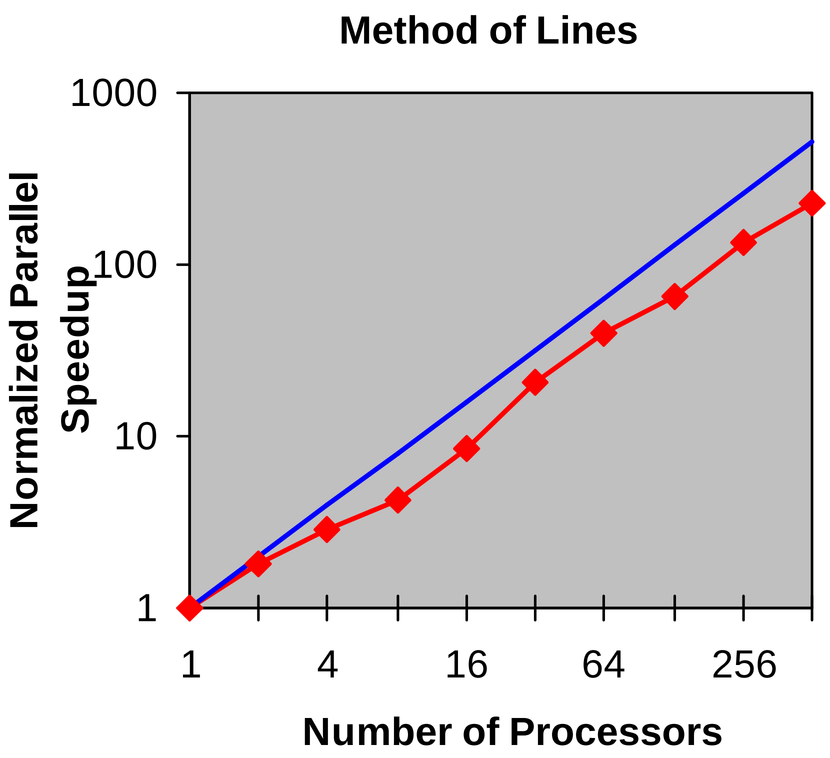
- SAMRAI introduction
- Parallel SAMR
- **Parallel performance measurements**
- New algorithms to enhance parallel performance
- Requirements and issues on BG/L

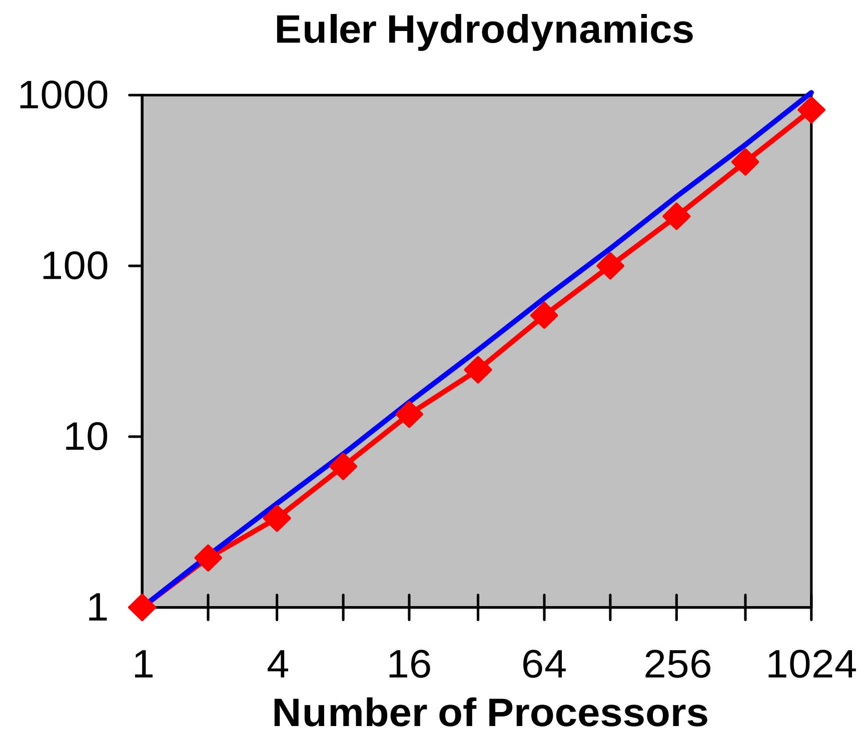
Non-adaptive calculations using SAMRAI show good scaling

Single Level calculation

50x50x50 patch per processor

IBM Blue Pacific





Sept 2000

Benchmarks constructed to analyze scaling properties of SAMR applications

- **Simple numerical kernels**
- **Invoke the main algorithmic components used in more complex apps (e.g. meshing, time advance, etc.)**
- **Timing decomposed into two phases:**

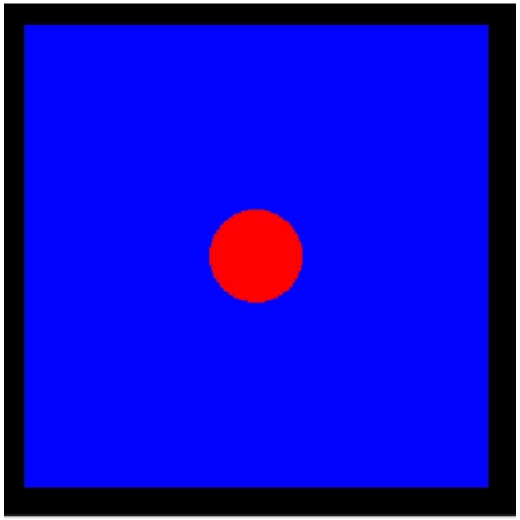
Time Advance:

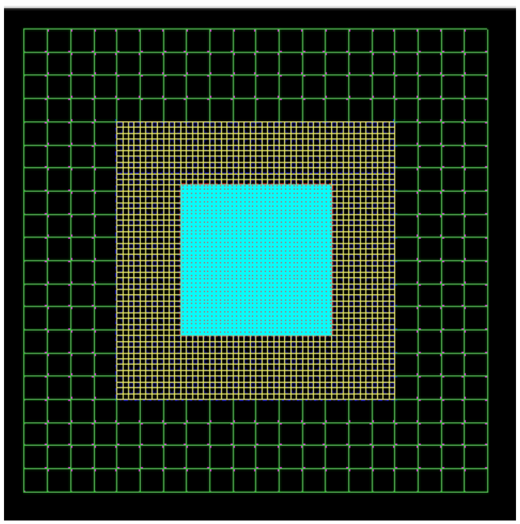
- numerical kernels
- communication (filling ghost cells)
- load imbalances

Re-Gridding:

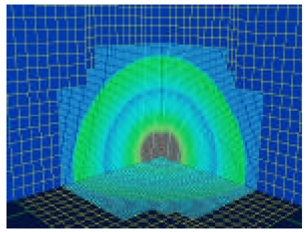
- cluster tagged cells
- construct communication schedules
- distributing data to new mesh configuration

Non-scaled Euler benchmark – same problem size run on all processors

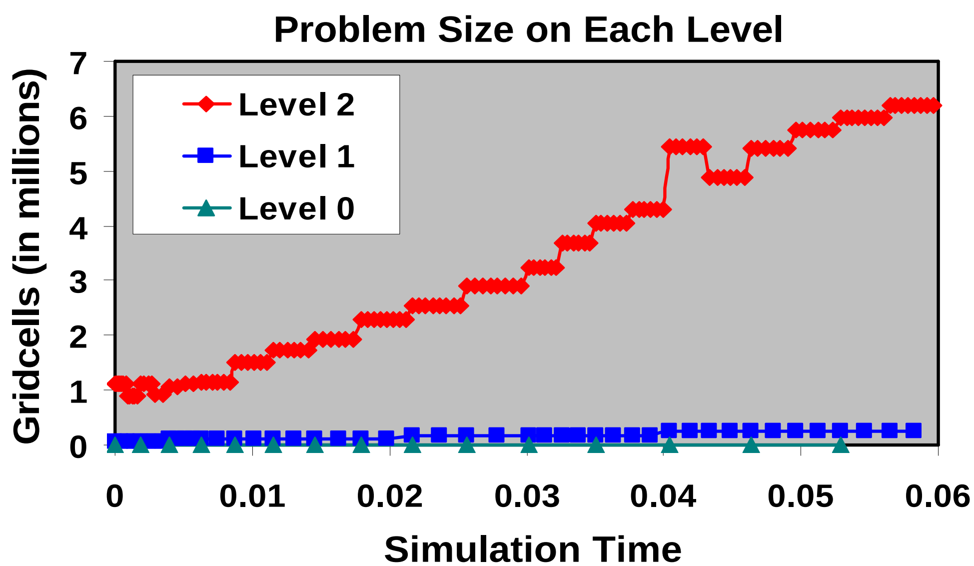




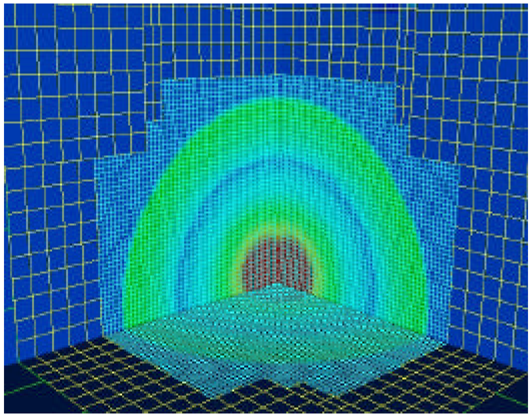
3D spherical shock - Euler hydrodynamics



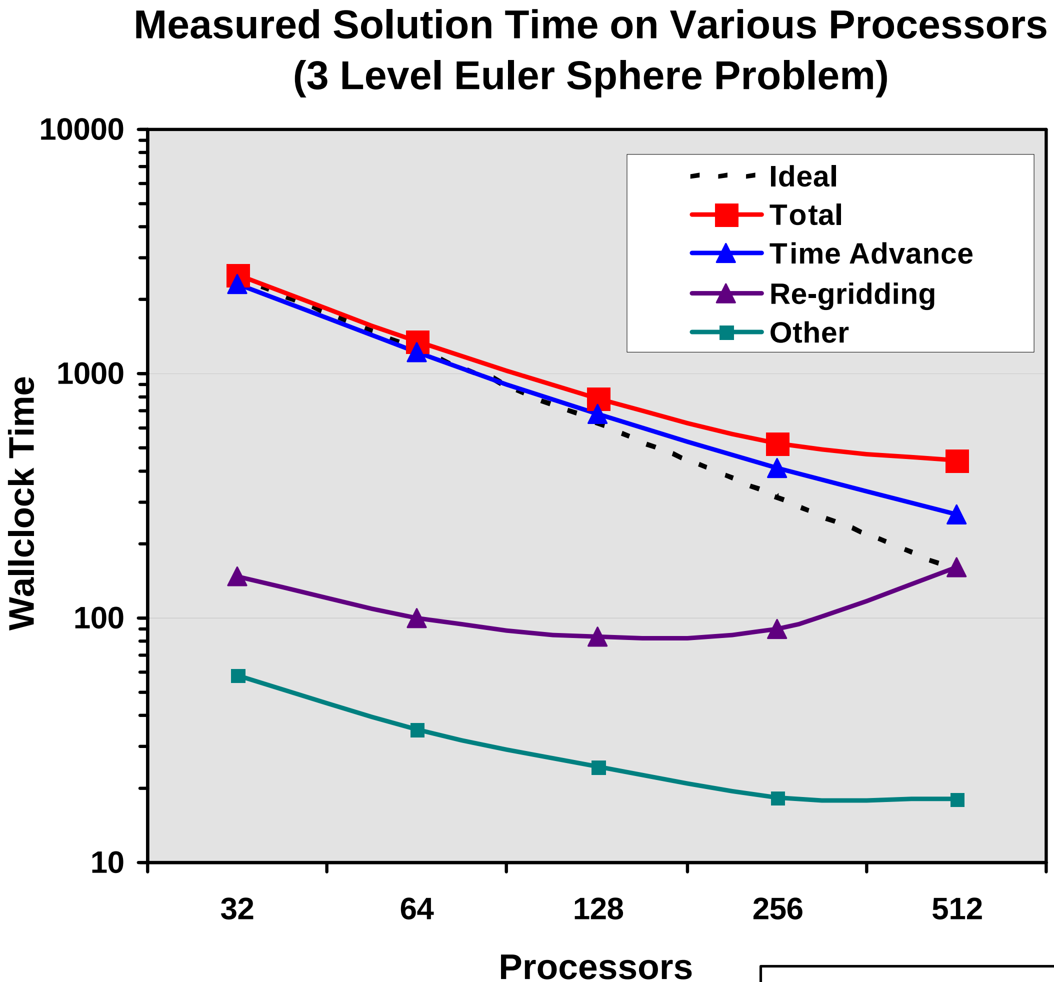
- Workload changes over simulation
- Per-processor workload decreases as number of processors increased



Parallel Performance of *non-scaled* adaptive Euler benchmark

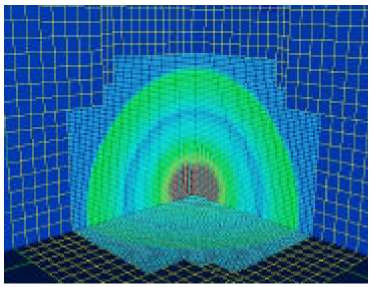


Non-scaled
Euler calculation
ASCI IBM Blue Pacific



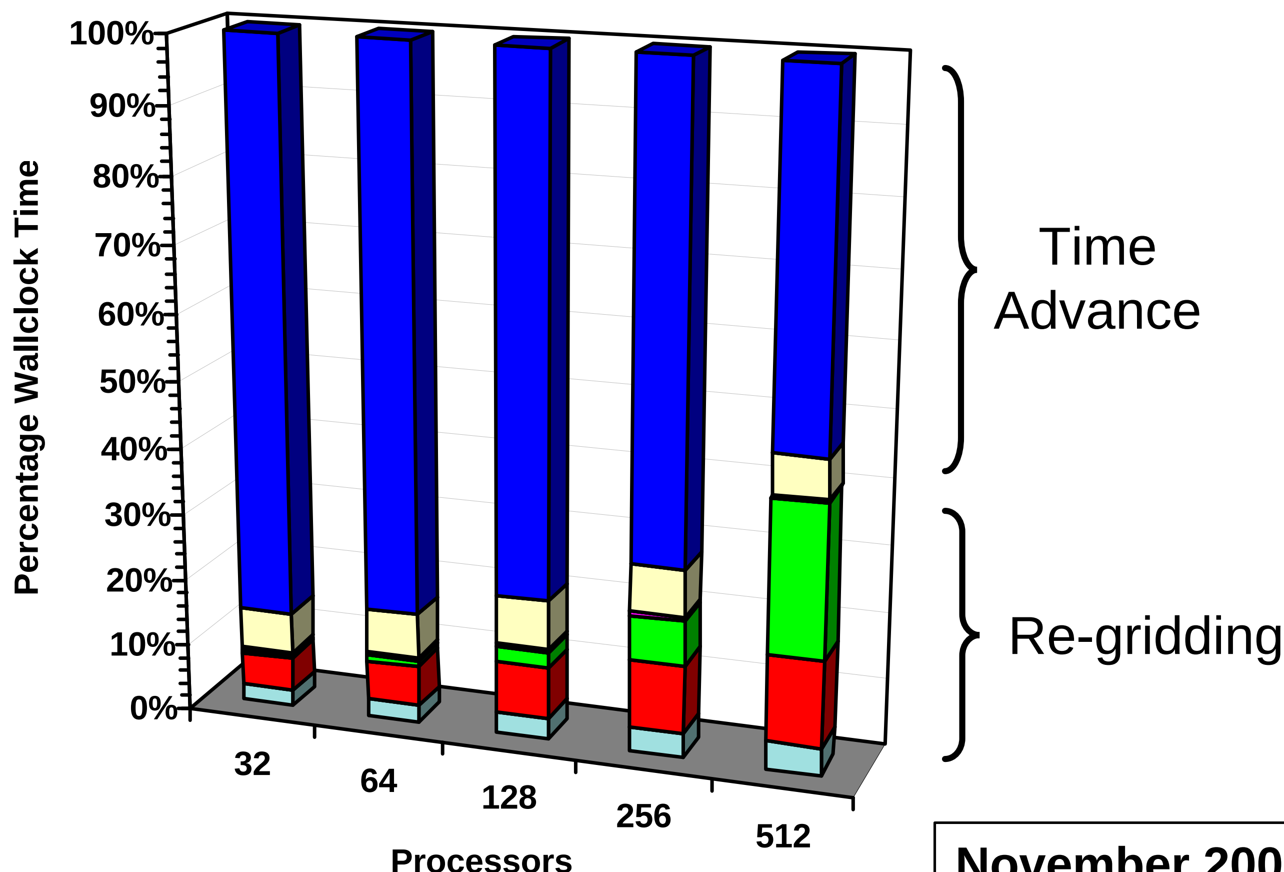
November 2001

Poor scaling in re-gridding hurts efficiency on large processor counts



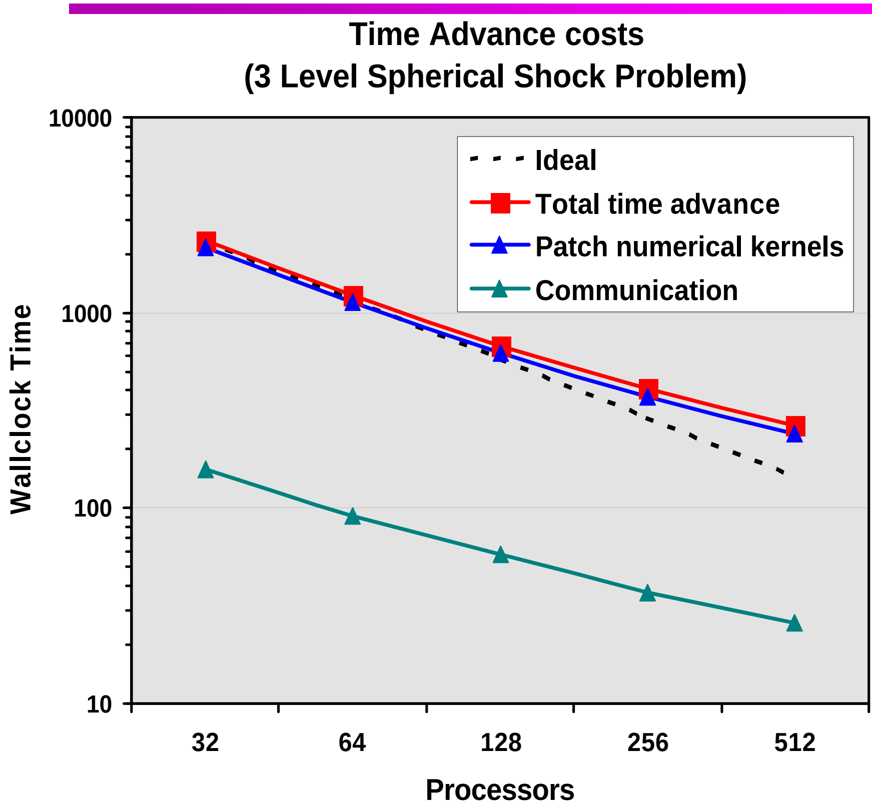
Non-scaled Euler calculation
ASCI IBM Blue Pacific

Measured Solution Time on Various Processors
(3 Level Spherical Shock Problem)

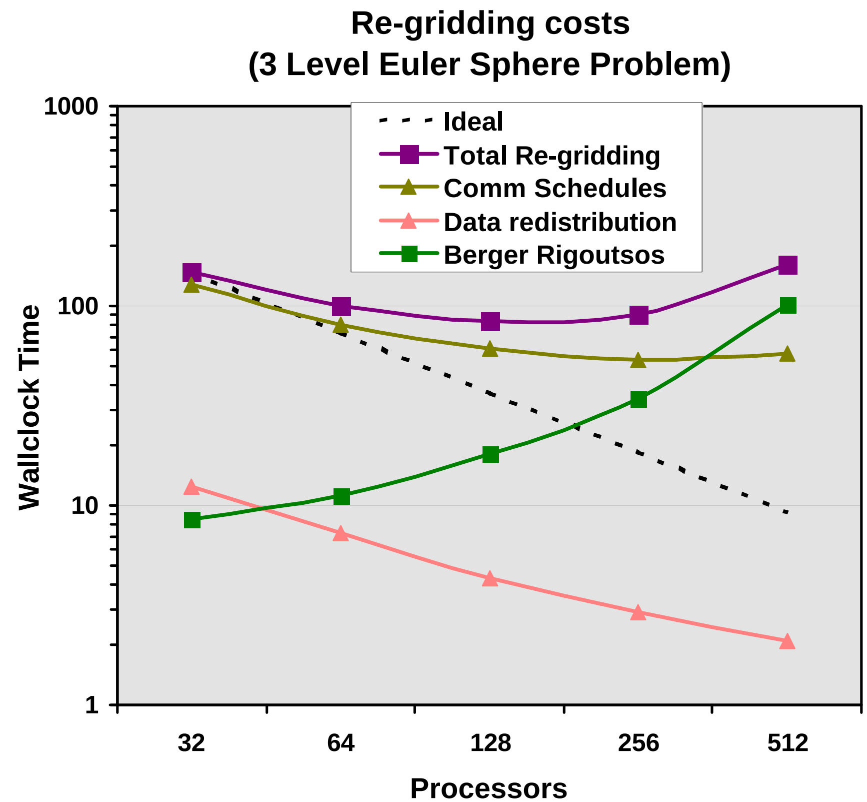


November 2001

Poor scaling in re-gridding hurts efficiency on large processor counts (ASCI Blue Pac)



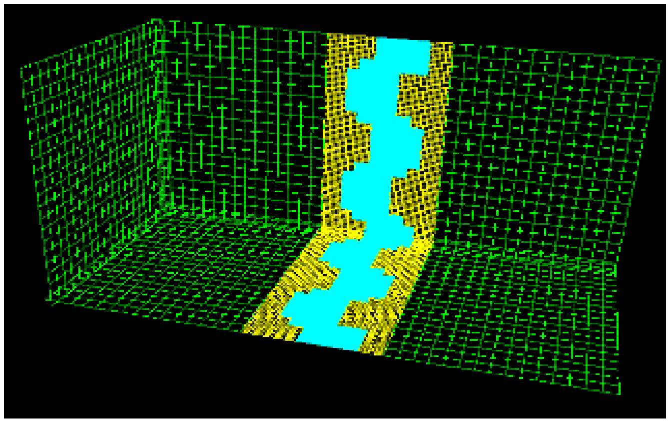
Operations performed while grid is fixed

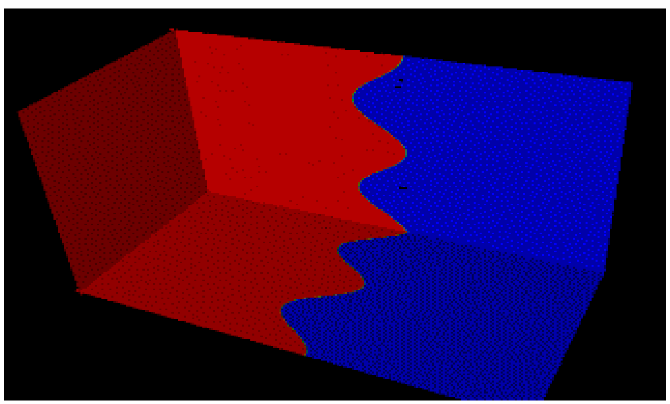


Re-gridding operations

November 2001

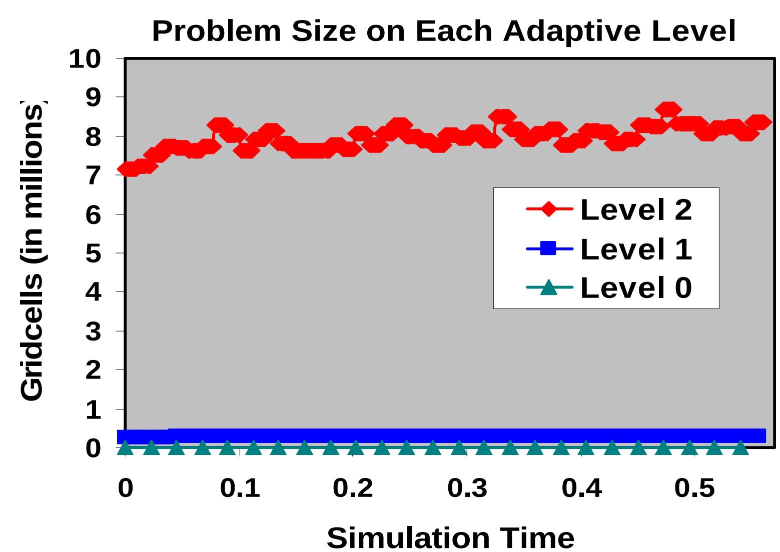
Scaled linear advection benchmark – problem size increased with processors



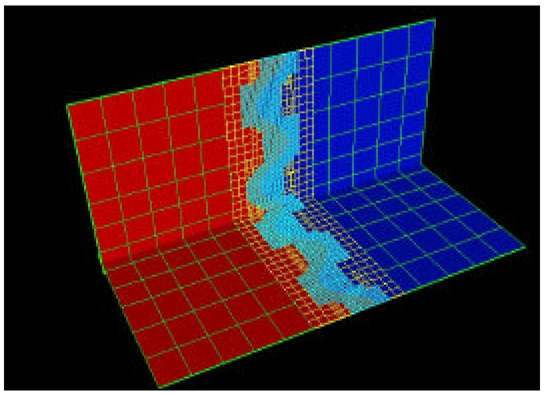


3D advecting sinusoidal front - linear advection

- Workload uniform over simulation
- Per-processor workload remains constant as number of processors is increased

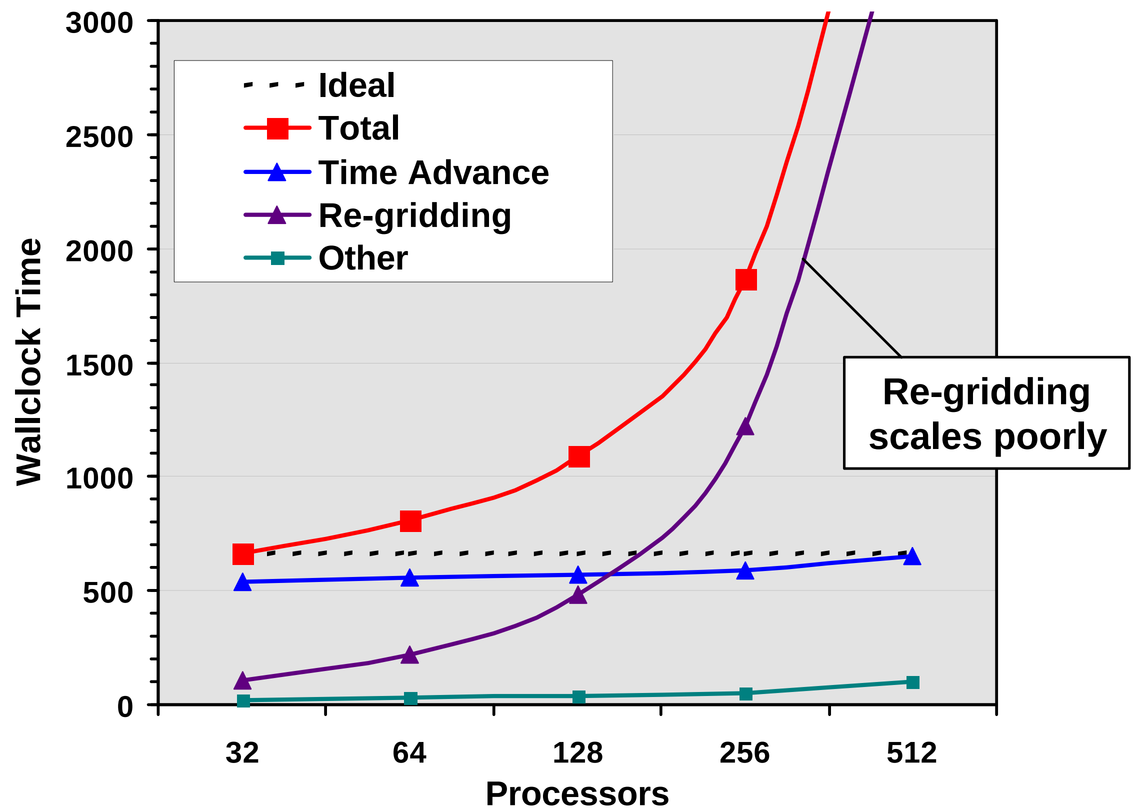


Parallel performance of scaled linear advection benchmark



Scaled
Linear advection calculation
ASCI IBM Blue Pacific

Measured Solution Time on Various Processors (3 Level Scaled Advecting Front Problem)



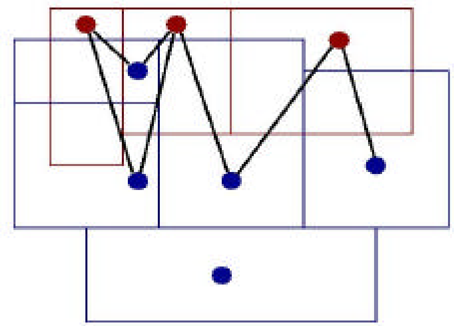
November 2001

Outline

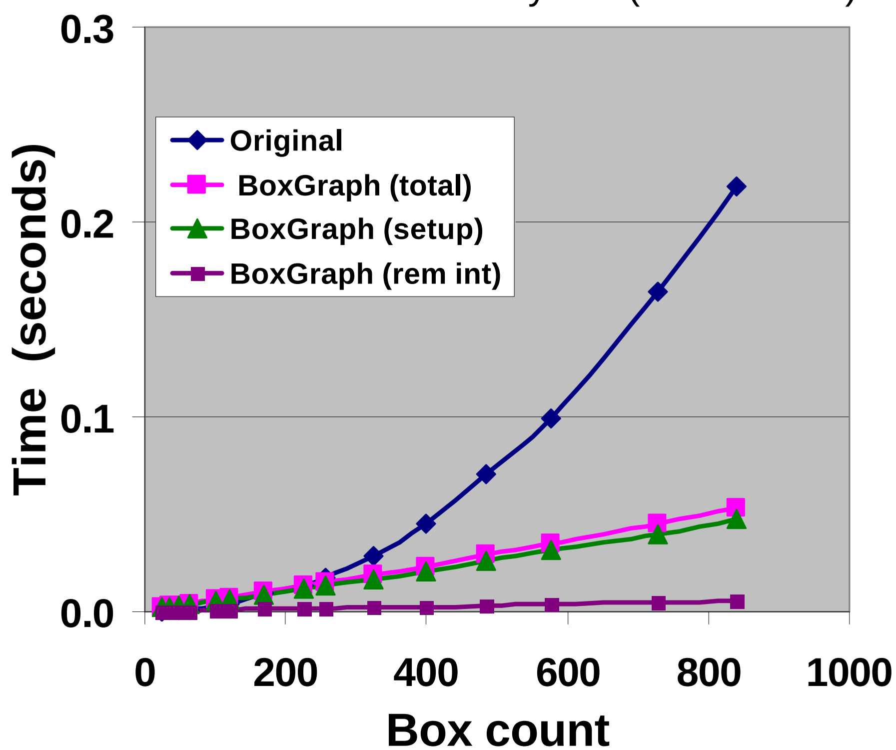
- SAMRAI introduction
- Parallel SAMR
- Parallel performance measurements
- **New algorithms to enhance parallel performance**
- Requirements and issues on BG/L

Graph-based algorithms speed up communication schedule construction

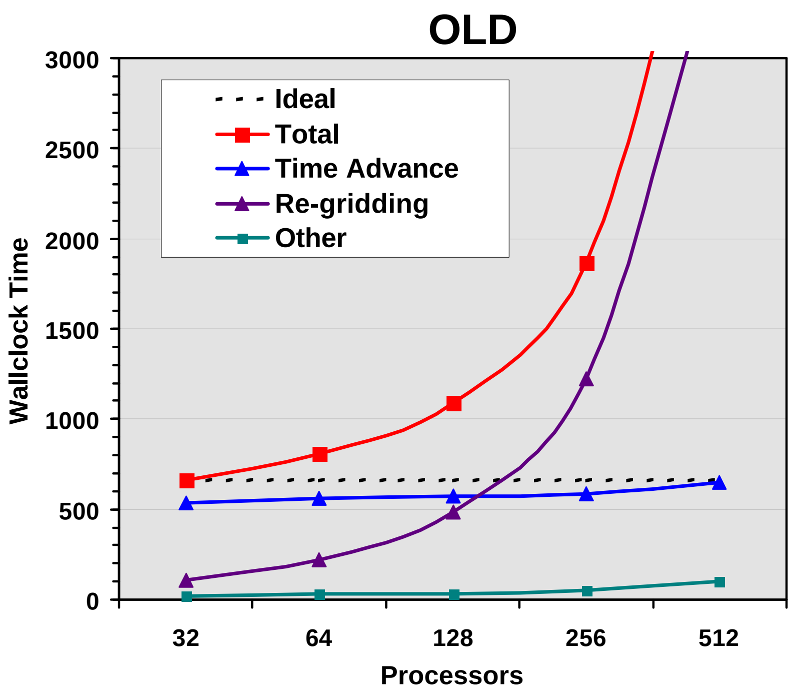
- Before constructing a communication schedule to transfer data between two levels, build a "Box graph":
  - Insert a vertex in *V* for each box
  - Insert an edge *(i,j)* at intersection
- Given this graph, can find a box's neighbors in O(1)
- Primary cost is graph construction

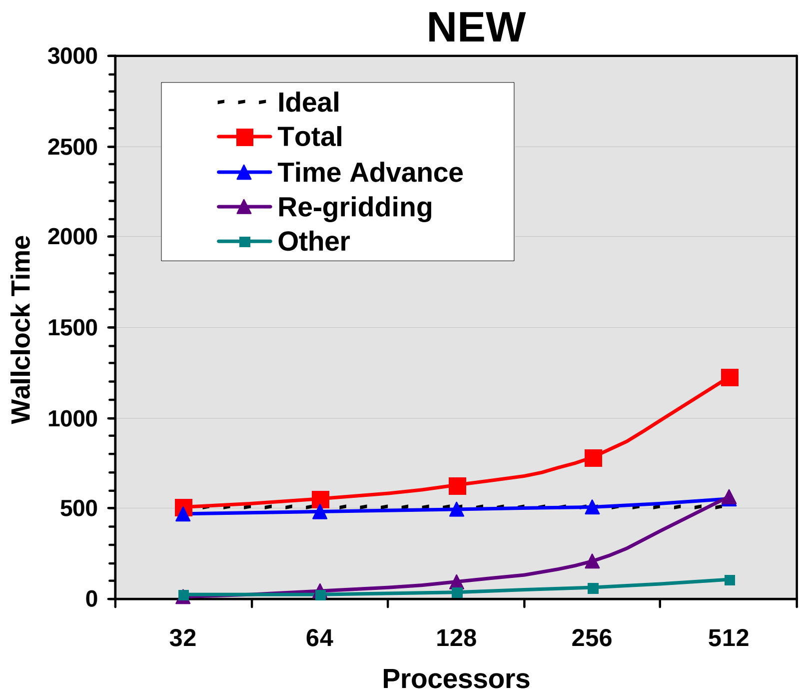


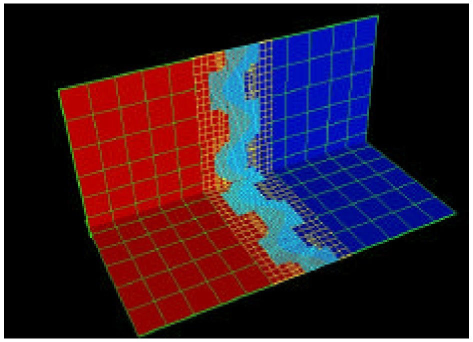
Hysom (CASC-LLNL)



Scaled results with new graph-based schedule construction algorithm



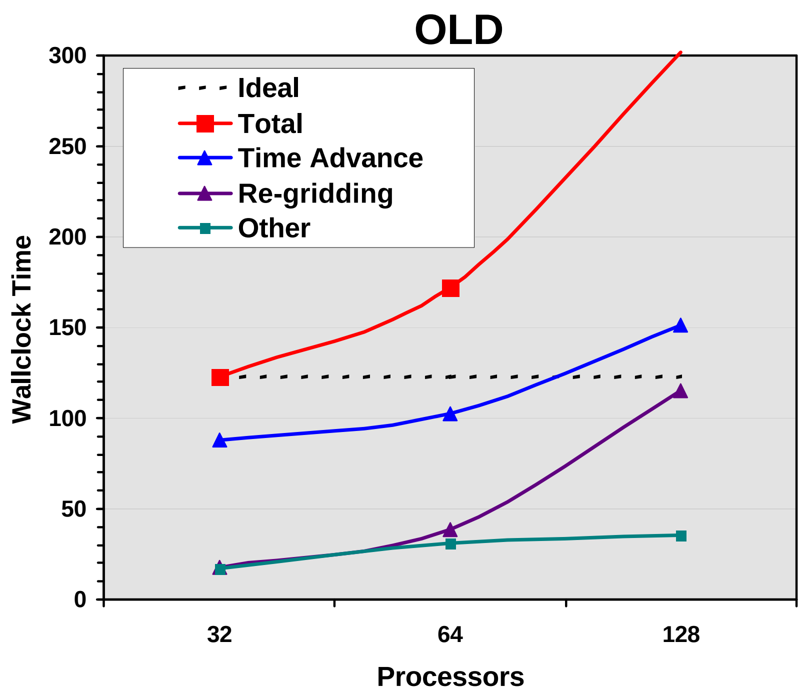


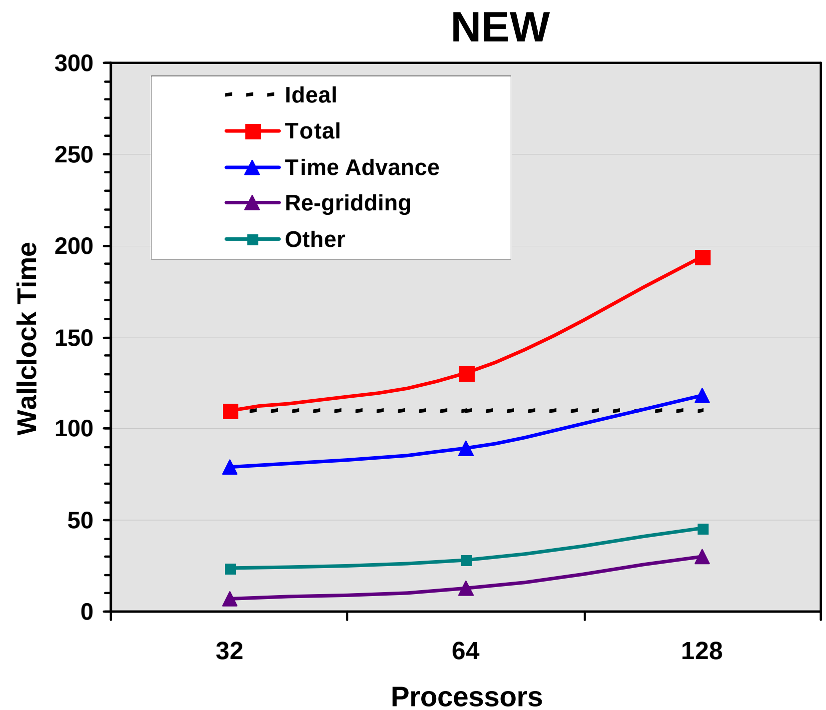


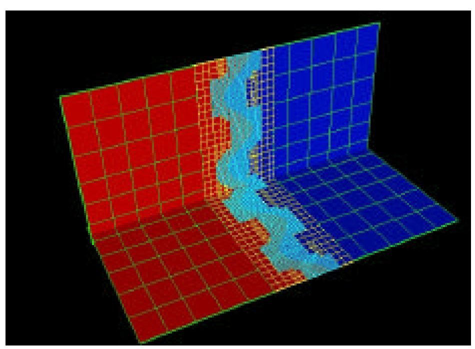
Scaled
Linear advection
calculation
IBM ASCI Blue Pacific

March 2002

Scaled results with new graph-based schedule construction algorithm





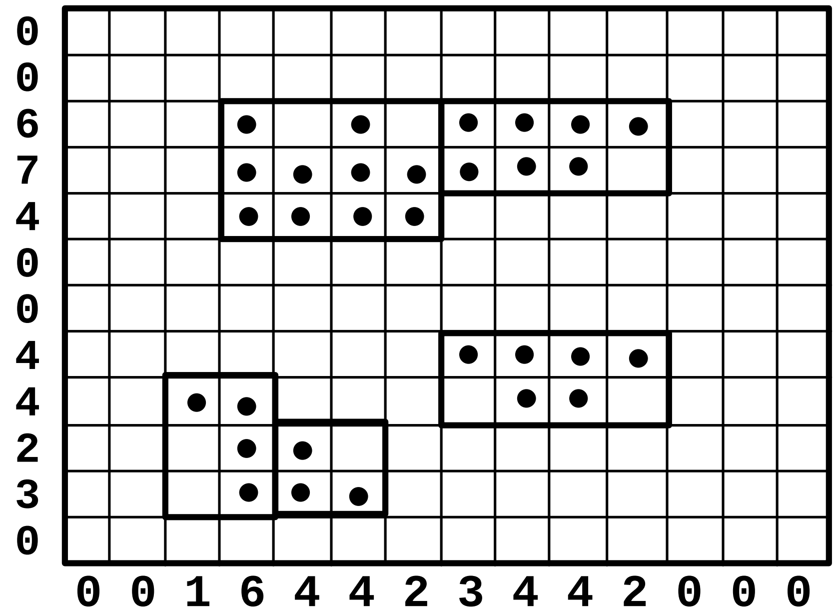


Scaled
Linear advection calculation
TC2K Compaq Cluster

March 2002

Binary Tree reduction for tagged-cell clustering algorithm (Berger-Rigoutsos)

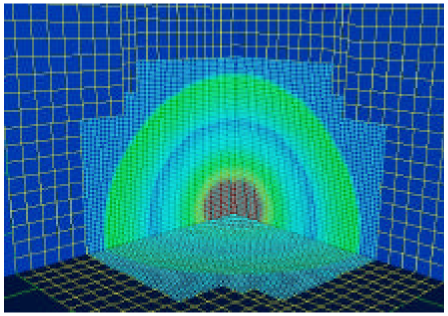
- **Berger-Rigoutsos:**
  - Forms new patches from tagged cells
  - Determines box cuts from global histogram through recursive algorithm
- **Original implementation used global reductions to form histogram**



- **New Implementation:**
  - **Binary-tree reduction** algorithm collects information from selected processors at each recursion.
  - New box configuration constructed and broadcast by one processor.

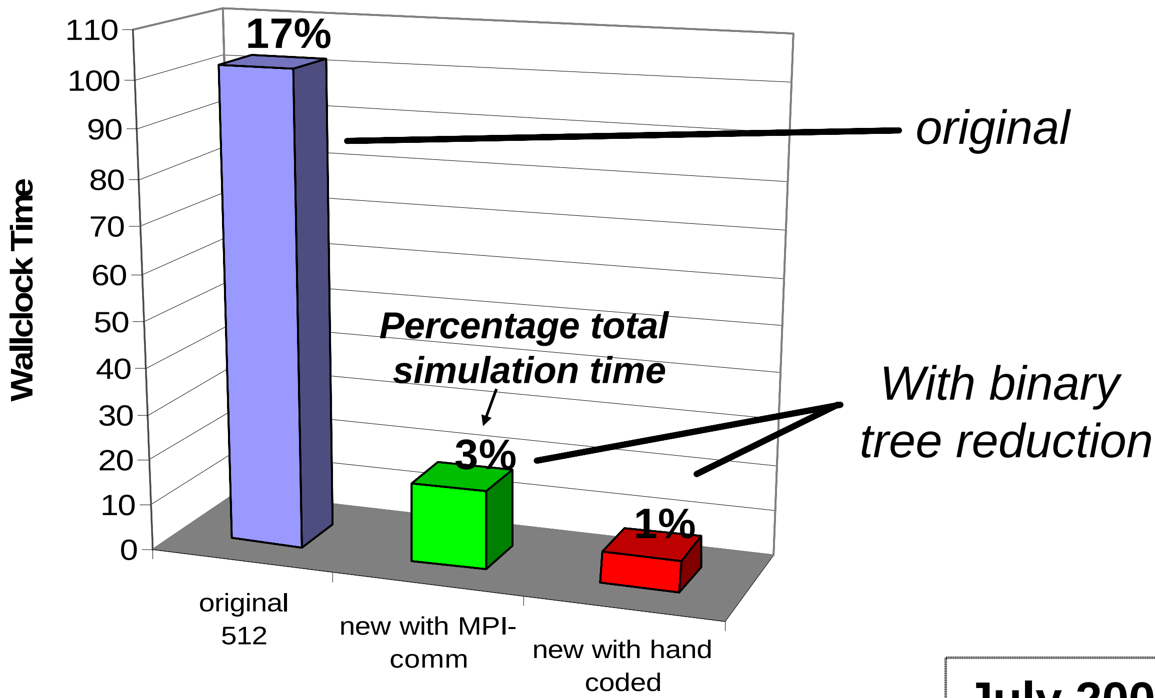
Timing results Berger-Rigoutsos algorithm with binary-tree reductions.

- **Binary tree reduction algorithm – two implementations**
  - MPI communicators
  - Hand-coded MPI send/recvs



Non-Scaled
Euler calculation
IBM ASCI Blue Pacific

Berger-Rigoutsos – 512 processors



July 2002

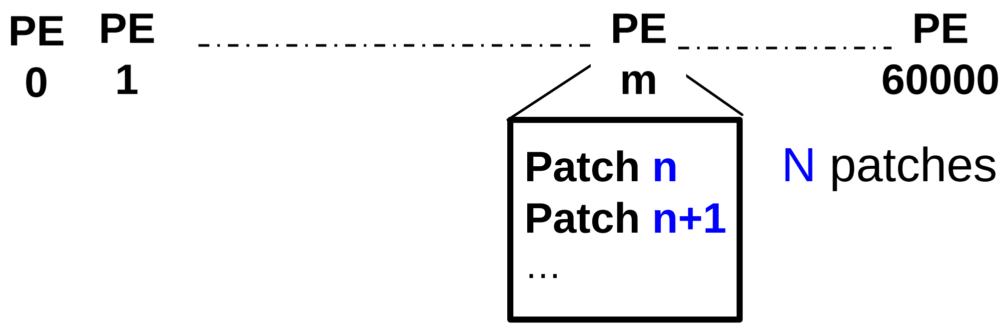
Outline

- SAMRAI introduction
- Parallel SAMR
- Parallel performance measurements
- New algorithms to enhance performance
- **Requirements and issues on BG/L**

BlueGene/L wish list

- **SAMRAI Dependencies:**
  - C++, C, F77/F90 compilers
  - MPI
  - HDF5 (checkpointing)
- **Desirable features:**
  - C++-capable debugging tool
  - Memory analysis tool (i.e. reports stack/heap usage on nodes)

Scaling issues with a large number of processors



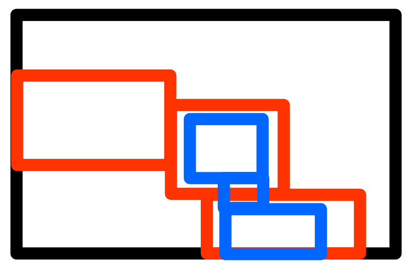
Number of patches scales with number of processors

- Box operations in gridding may invoke $O(N^2)$ algorithms (e.g. former communication schedule algorithm).
- More efficient graph-based algorithms work on up to 512 processors, but ***need to develop efficient algorithms for O(10K) processors***.
- Difficult to assess beforehand because complexity is generally problem dependent.

BG/L will require us to rethink our grid storage approach

- **Current approach: Each MPI process holds a box for each patch in the problem to determine communication dependencies.**

`level->getBoxes();`



- **Because # patches grows with # processors, *trivial overhead becomes non-trivial on BG/L.***

procs	patches	per processor storage (MB)
0.5K	2.5K-10K	< 1 MB
60K	300K-1200K	20-80MB

Large overhead for nodes of BG/L

Collective communications on BG/L

- **Berger-Rigoutsos clustering:**
  - **Binary tree reduction algorithm effective in reducing costs on O(0.5K) processors.**
  - **Will this approach be effective on O(10K) processors?**
- ***Some* global communications are necessary (e.g. timestep synchronization in time advance).**

Concluding Remarks

- **Porting SAMRAI to BG/L enables a variety of applications to use the architecture.**
- **Results of scaling AMR algorithms on up to 512 processors:**
  - Communication *not* the primary source of scaling inefficiency.
  - Re-gridding operations that are trivial on small numbers of processors become significant on large numbers.
  - More efficient graph-based algorithms successful in reducing these costs.
- **Speculation on running AMR applications on BG/L:**
  - Re-gridding costs will likely be the main hindrance.
  - Continued exploration into more efficient gridding algorithms needed.

Auspices Statement

- This work was performed under the auspices of the U.S. Department of Energy by University of California Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48.

- Document UCRL-PRES-149437